LLM Interpretability

(Mechanistic interpretability)

(reproducing and extending "Scaling Monosemanticity" by Anthropic)

Drejc Pesjak

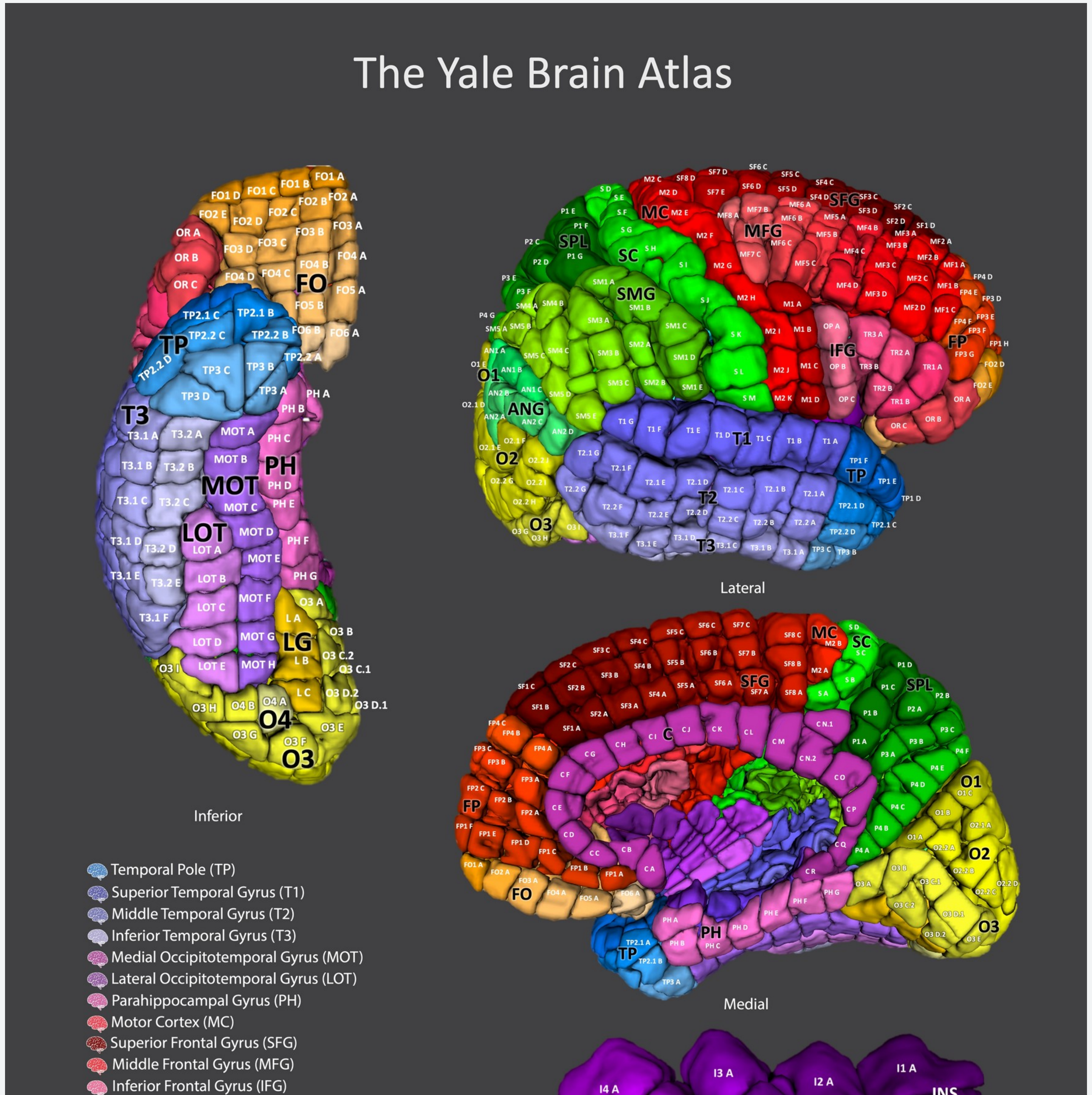
Under the mentorship of Jan Rupnik

Jožef Stefan Institute

Department for Artificial Intelligence

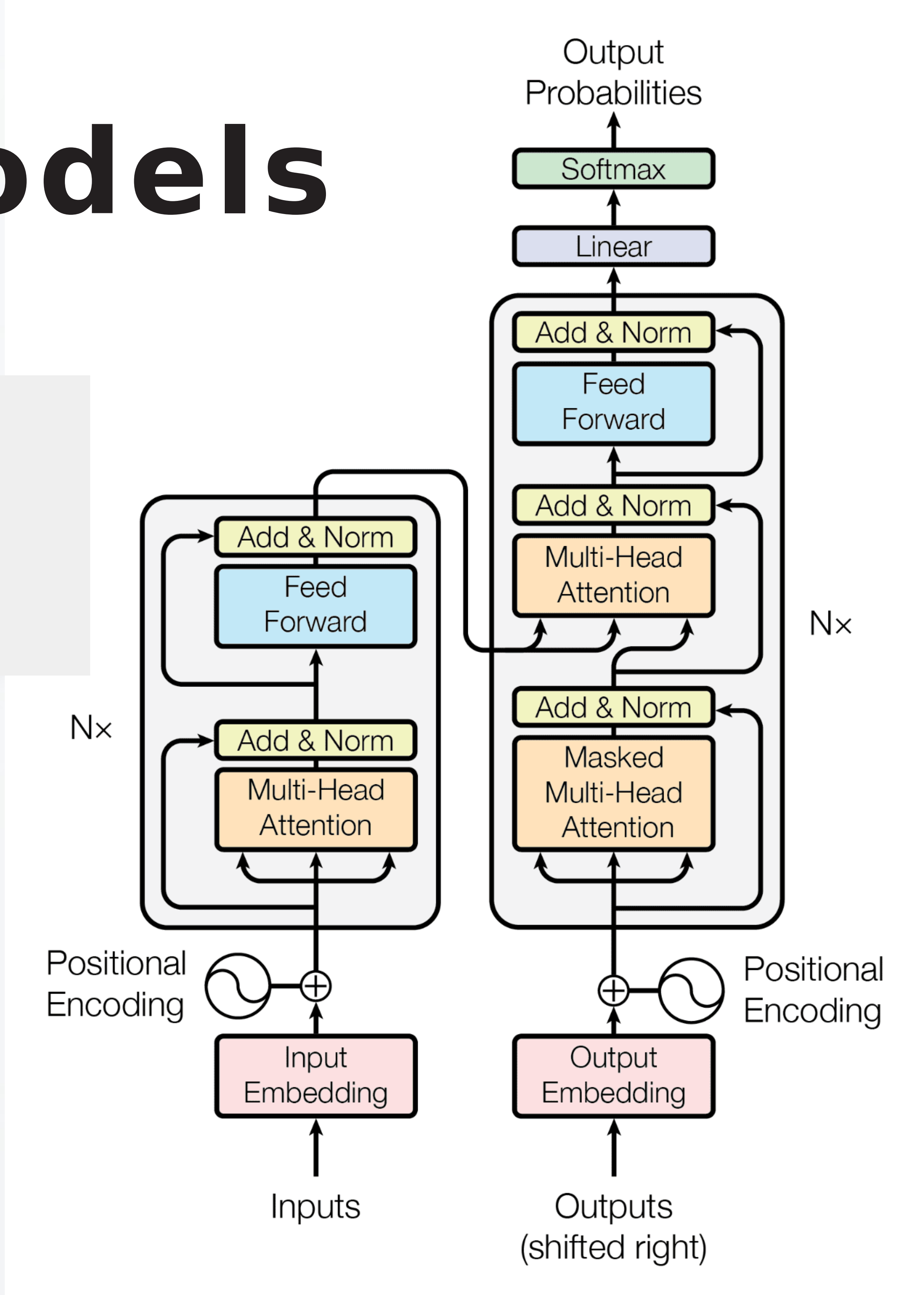
ET LLM 2025 DOJO

Motivation

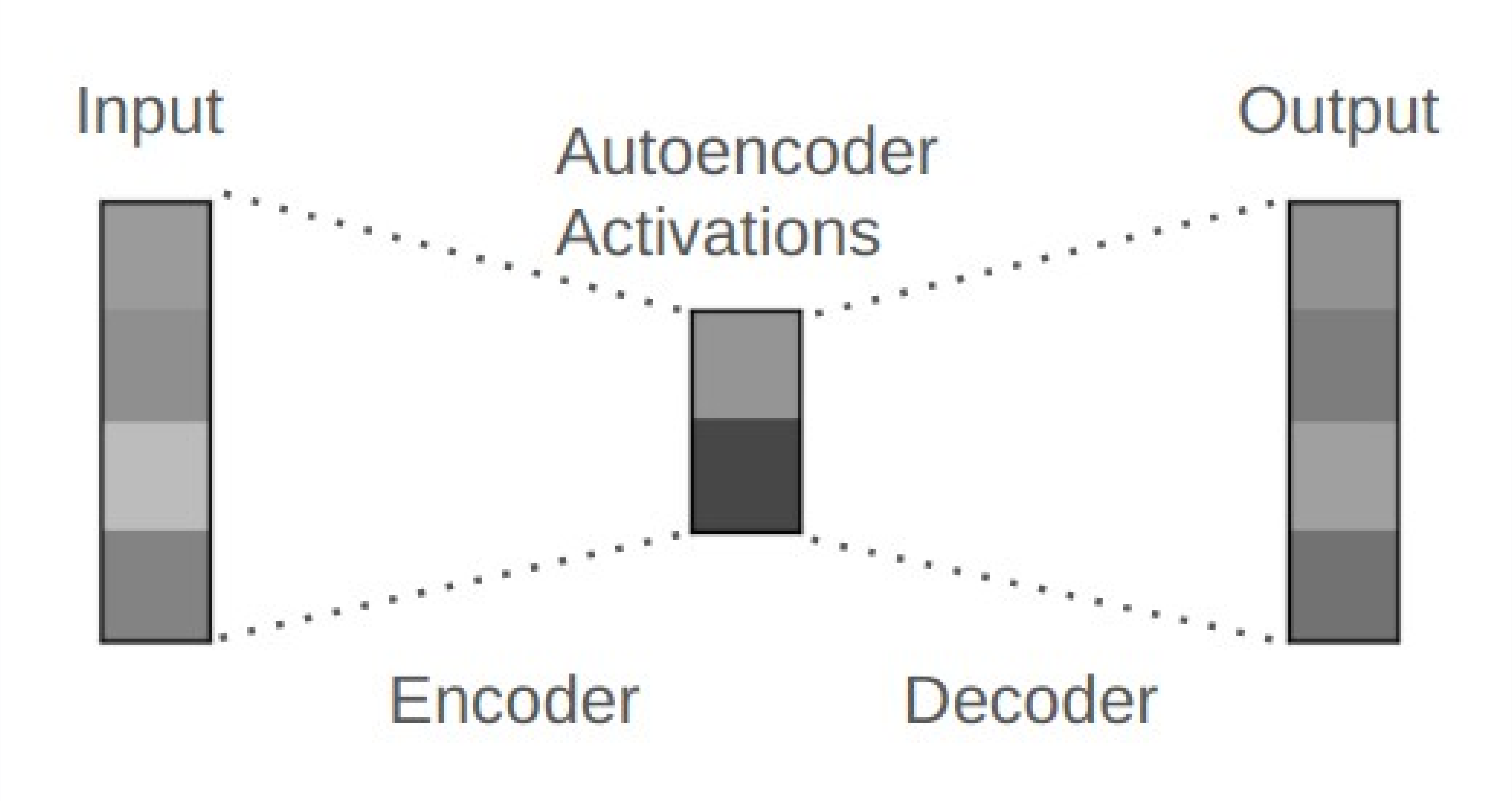

The Yale Brain Atlas
Inferior
Lateral
Medial
Temporal Pole (TP)
Superior Temporal Gyrus (T1)
Middle Temporal Gyrus (T2)
Inferior Temporal Gyrus (T3)
Medial Occipitotemporal Gyrus (MOT)
Lateral Occipitotemporal Gyrus (LOT)
Parahippocampal Gyrus (PH)
Motor Cortex (MC)
Superior Frontal Gyrus (SFG)
Middle Frontal Gyrus (MFG)
Inferior Frontal Gyrus (IFG)

Large Language Models

- LLaMa 3.2 3B
- Huggingface
- 4-bit quantized
- 16^{th} layer



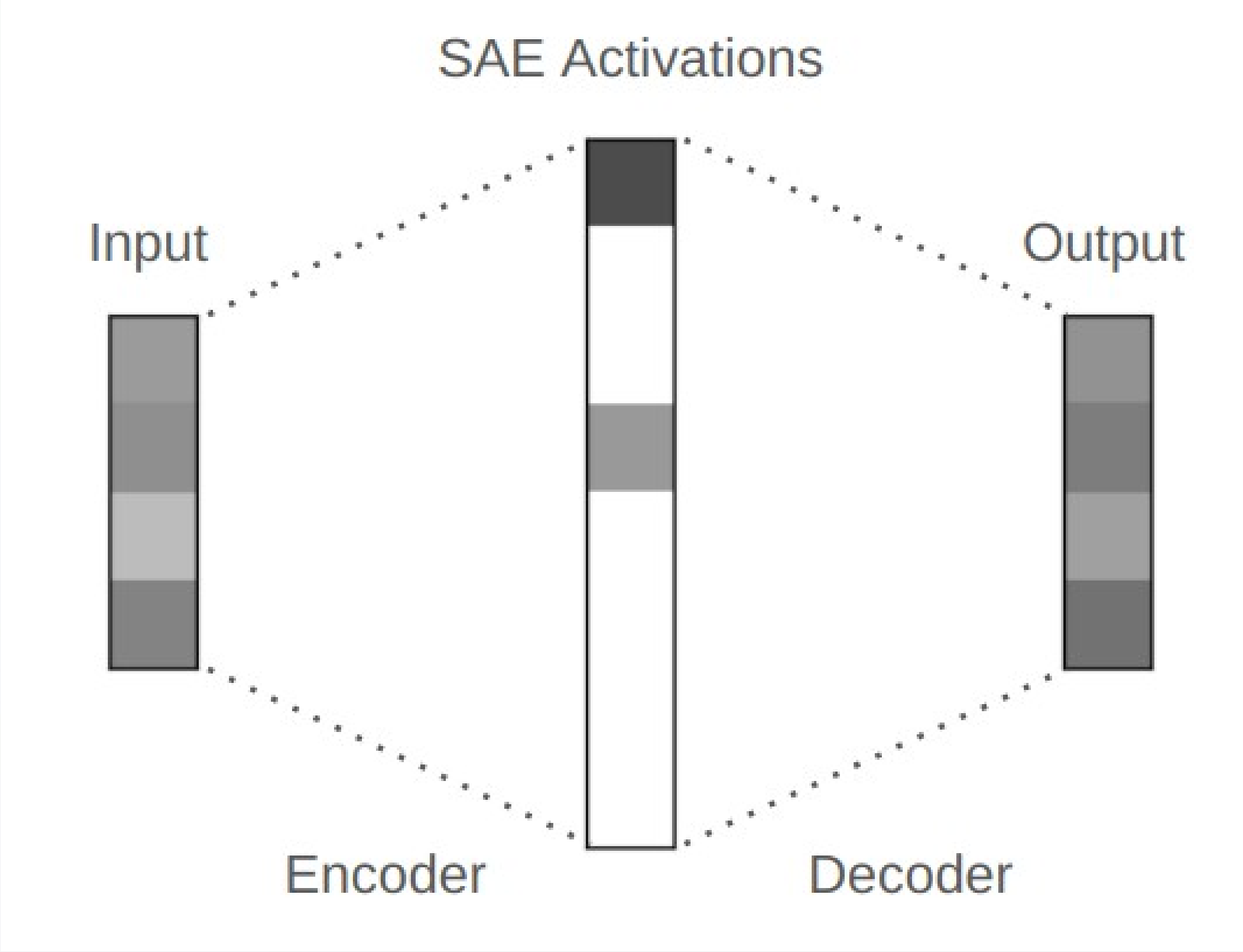
Autoencoder



https://adamkarvonen.github.io/machine_learning/2024/06/11/sae-intuitions.html

Sparse autoencoder

(overcomplete)



Sparse autoencoder

(overcomplete)

```python
class SparseAutoencoder(nn.Module):
    def __init__(self, input_dim, hidden_dim):
        super(SparseAutoencoder, self).__init__()
        # Encoder
        self.encoder = nn.Linear(input_dim, hidden_dim)
        # Decoder
        self.decoder = nn.Linear(hidden_dim, input_dim)

    def forward(self, x):
        encoded = torch.relu(self.encoder(x))
        decoded = self.decoder(encoded)
        return decoded, encoded
```

Sparse autoencoder

(overcomplete)

```python
# Set up loss function and optimizer
criterion = nn.MSELoss().to(device)
optimizer = optim.Adam(model.parameters(), lr=0.0001)
l1_lambda = 0.01  # Regularization strength for sparsity

for epoch in range(num_epochs):
    total_loss = 0
    for i, batch in enumerate(data_loader):
        print("Batch number: ", i)
        # Forward pass
        batch = batch.to(device)
        outputs, encoded = model(batch)
        mse_loss = criterion(outputs, batch)

        # Add L1 regularization for sparsity
        decoder_weight_norms = torch.norm(model.decoder.weight, p=2, dim=0)
        l1_terms = encoded * decoder_weight_norms.unsqueeze(0)
        l1_loss_per_sample = torch.sum(l1_terms, dim=1)
        l1_loss = torch.mean(l1_loss_per_sample)

        loss = mse_loss + l1_lambda * l1_loss

        # Backward pass and optimization
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}} \left[\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \lambda \sum_i f_i(\mathbf{x}) \cdot \|\mathbf{W}_{:,i}^{dec}\|_2 \right]$$

Training SAE

Kaggle

- Pytorch lightning and Ray tune – hyperparameter tuning
- Kaggle free 30h/week P100 gpu
- Ran about 100 tests

- Best model:

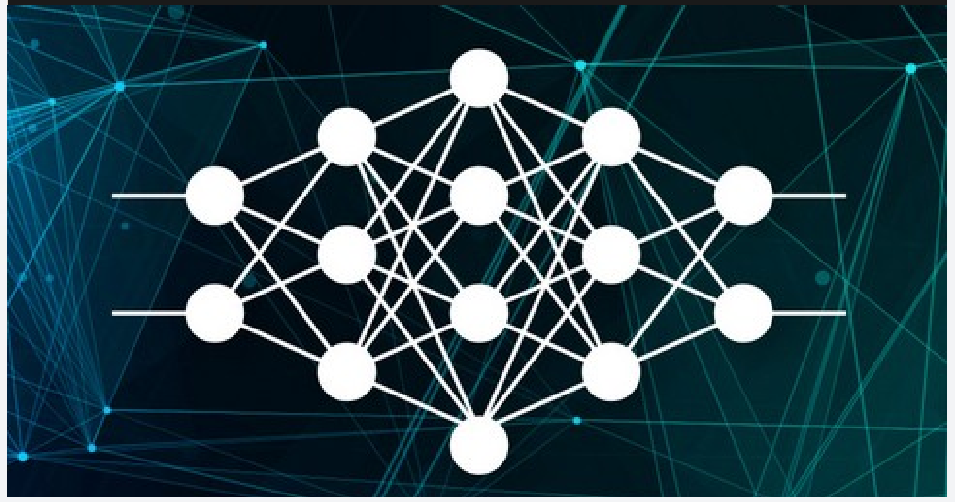
  'input_dim': 3072,

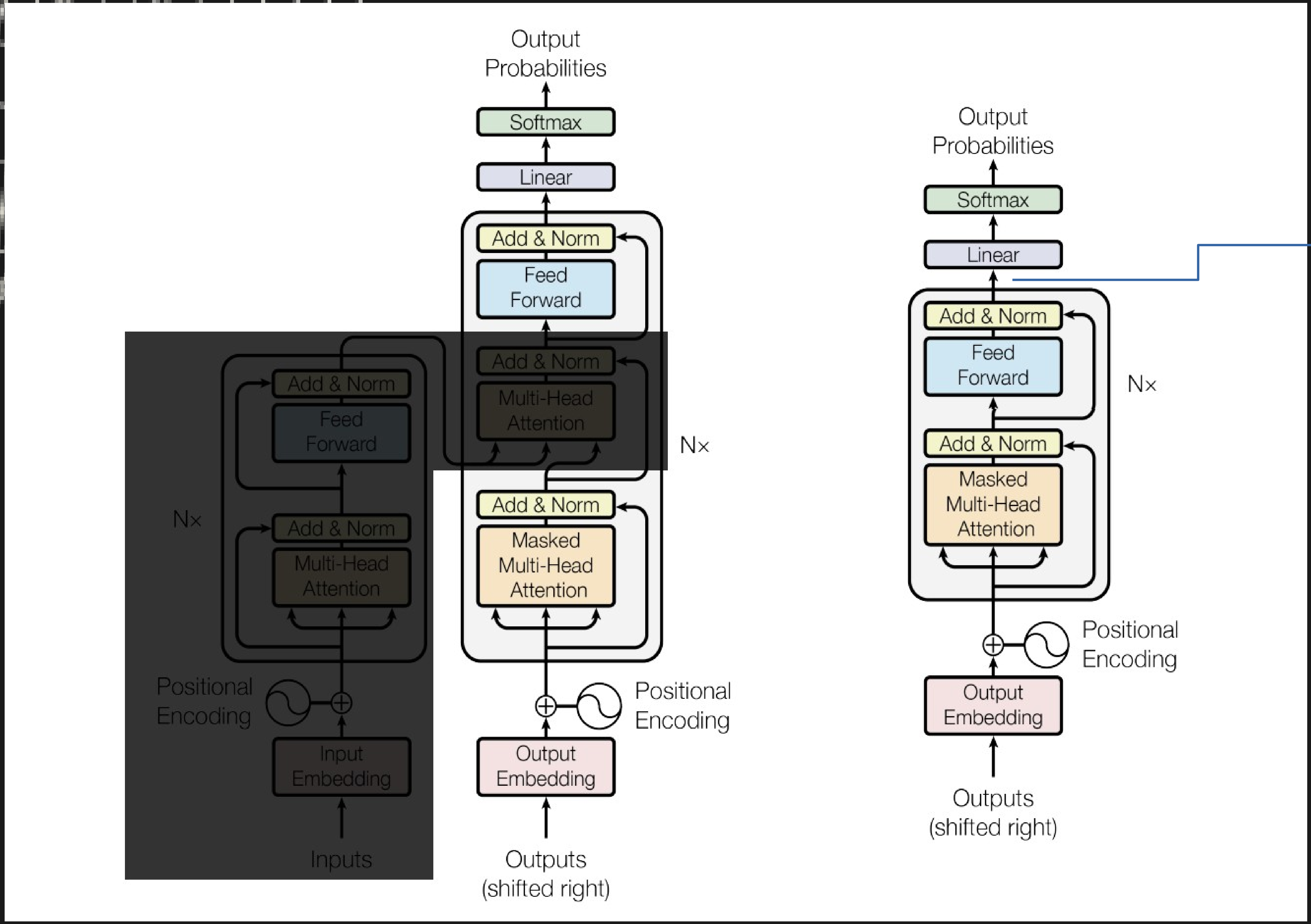
  'hidden_dim': 65536,

  'l1_lambda': 0.00597965,

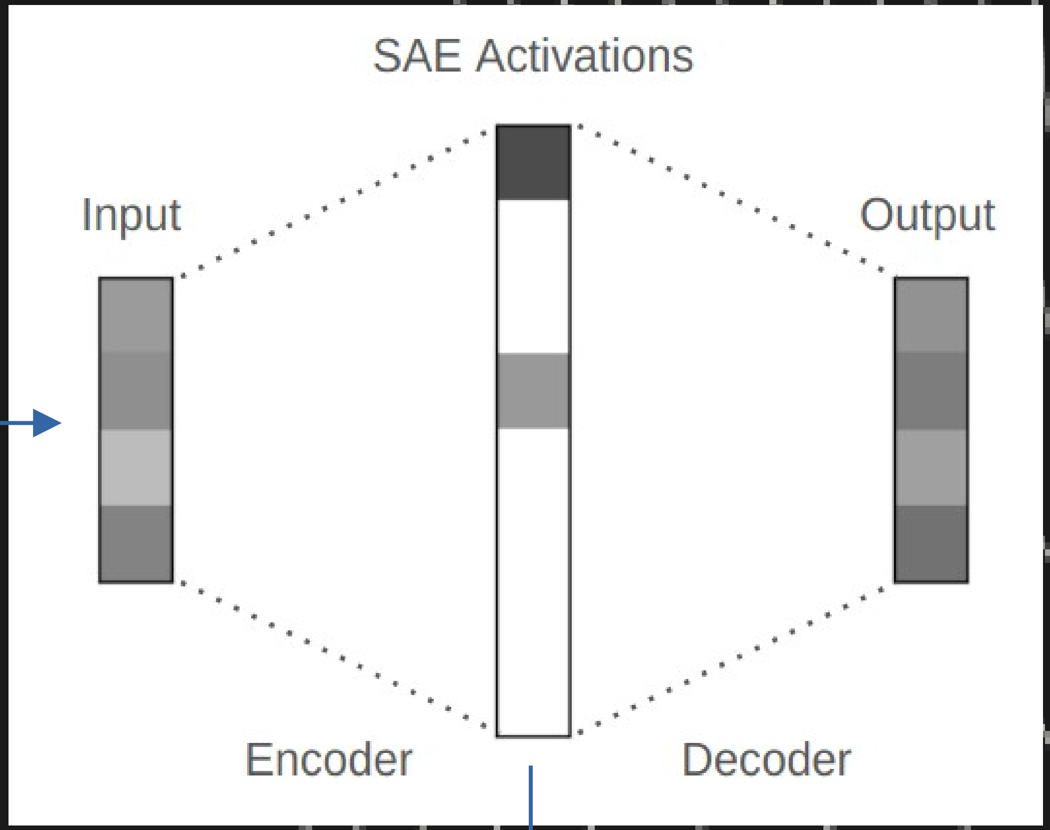
  'lr': 2.5011e-05

Best





https://www.ericjwang.com/2023/01/22/transformers.html



ANALYZE

Analysis - interpretability

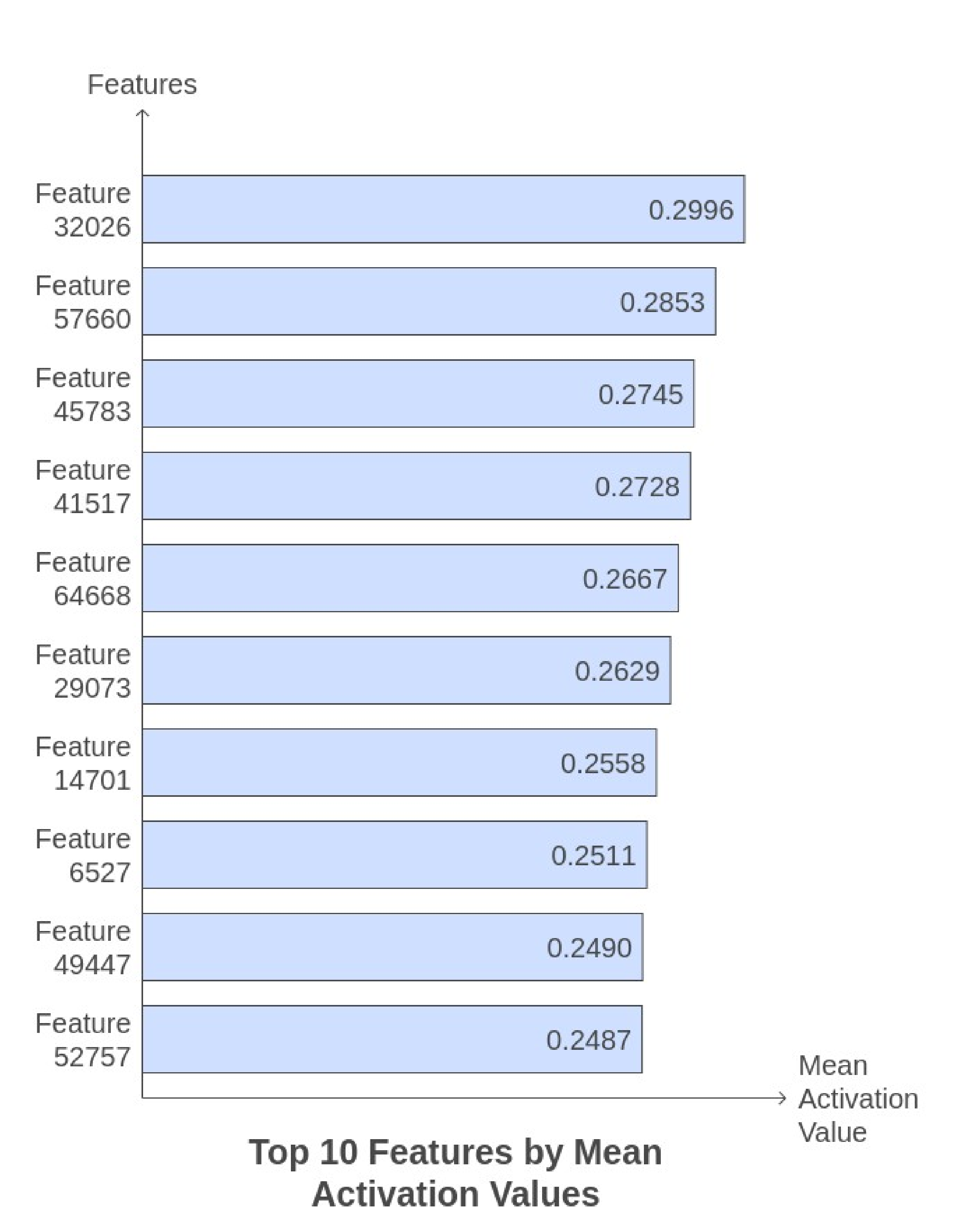
Prompt search

- program on aerobic capacity and muscle strength of adults with hearing loss. Twenty-three adults with hearing loss were separated into 2 groups. Thirteen subjects
- the effect of a traditional dance training program on aerobic capacity and muscle strength of adults with hearing loss. Twenty-three adults with hearing loss were separated into
- been examined comprehensively. Peritoneal lavage was performed in 351 patients before curative resection of a gastric carcinoma between 1987 and

- Tokenized input prompts
- Processed through the LLM model
- Representations extracted from the 16th layer
- Passed through the SAE encoder
- Output: latent sparse vectors

Analysis - interpretability

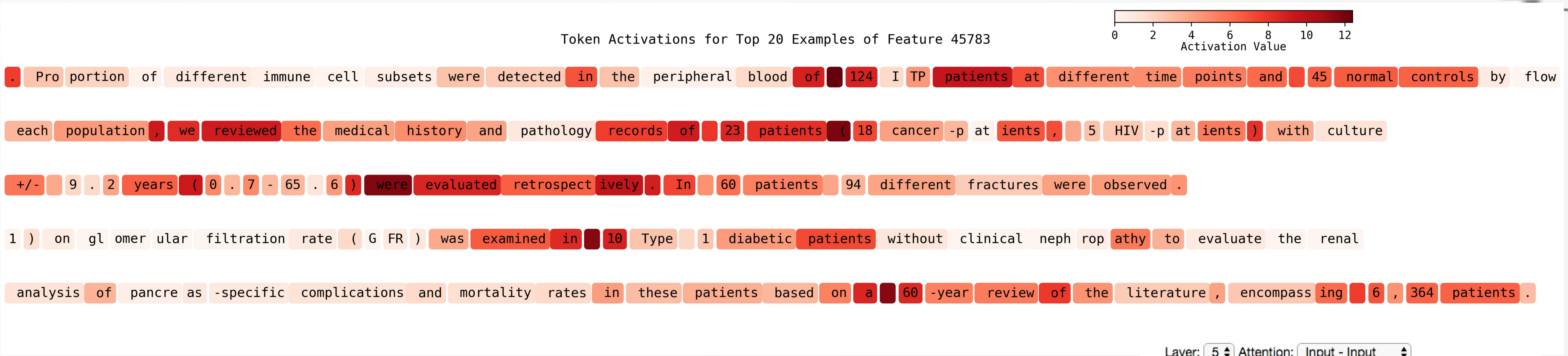
Feature retrieval

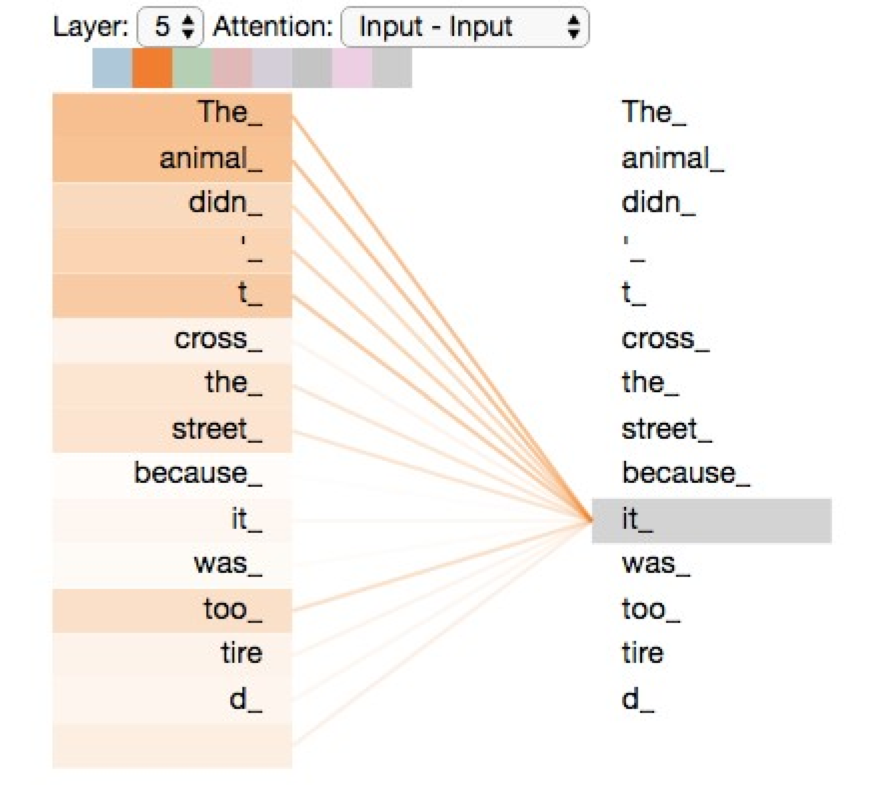


- Why mean and not max?

Analysis - interpretability

TopK examples





https://jalammar.github.io/illustrated-transformer/

Analysis - interpretability

Automatic feature explanation

- ChatGPT-4o (max input length 32k tokens)
- LLaMa 3.2 3B (Ollama) – works just as fine
- Input = prompt + topK examples
- Output:

Feature Index [45783]

Dominant Tokens: 'patients', '(', 'Fifty', ';', 'into'

Patterns: Activates in medical or clinical study contexts, often quantifying patients or describing study methodologies.

Summary: Highlights patient-focused data or study details in medical literature.

Context: Found in detailed descriptions of clinical trials or patient demographics.

Title: Clinical Study Patients

Analysis - interpretability

Automatic feature explanation

1. **Feature 32026**: Scientific Study Purpose
2. **Feature 57660**: Academic References
3. **Feature 45783**: Clinical Study Patients
4. **Feature 41517**: Experiment Validation
5. **Feature 64668**: Action and Roles
6. **Feature 29073**: Conversational Context
7. **Feature 14701**: Quantitative Demographics
8. **Feature 6527**: Population Studies
9. **Feature 49447**: Technical Problem-Solving
10. **Feature 52757**: Medical Study Terms

Analysis - interpretability

Influence (steering)

- Starting prompt:
  - I am a
- Zero Boost:
  - I am a little confused about the meaning of the word ‘sociology’ in the title of this book. I have read the book and I am not sure what the word ‘s
- 30x Boost on Feature 45783:
  - I am a 20 year old female who has been diagnosed with a rare disease called SLE (systemic lupus erythematosus) and have been diagnosed with 3 cases of pulmonary

- Adjusting the “personality” of LLM
  - Without training/finetuning/RLHF or prompting
  - Make it nice/friendly, without biases, truthful...

Use case - deception

- During inference monitor the “lying” feature

Next steps

- Bigger and better
  - LLM, SAE, dataset
  - TopK SAE

Everest Cantu

THANK YOU

https://github.com/DrejcPesjak/scaling-monosemanticity-llama